JOURNAL OF COMPUTER AND SYSTEM SCIENCES 27, 77–100 (1983)

On the Complexity of Chess

JAMES A. STORER*

Bell Laboratories, Murray Hill, New Jersey 07974

Received June 29, 1979; revised December 12, 1980

It is shown that for any reasonable generalization of chess to an *NxN* board, deciding for a given position which player has a winning strategy it is PSPACE-complete.

1. INTRODUCTION

Most past work analyzing games from the point of view of computational complexity has dealt with combinatorial games on graphs (e.g., Even and Tarjan [3], Schaefer [10], Chandra and Stockmeyer [2]). However, recently Fraenkel *et al.* [5], and Lichtenstein and Sipser [8] have considered the game of checkers and GO, respectively. These authors show that for generalizations of checkers and GO to an *NxN* board, it is PSPACE-hard[1] to determine if a specified player has a winning strategy. This paper shows that for a wide class of generalizations of chess to an *NxN* board, it is PSPACE-complete to determine if a specified player has a winning strategy.

In order to study a game like checkers, GO, or chass in terms of asymptotic complexity, it is clearly necessary to generalize the game to an *NxN* board. With GO this is no problem, since the starting position is the empty board and all pieces are the same. To generalize checkers, a starting position must be specified, but since all the pieces are the same, this is not a great difficulty, and the PSPACE-hardness of checkers was shown for virtually any reasonable starting position. With chess, however, it is not clear how to generalize. One can imagine all sorts of new pieces and a host of starting positions. In addition, the knight of standard chess poses a problem. It seems reasonable that generalized chess would have at least one of the standard queen, rook, and black and white bishops. However, as the board size gets large, standard knights seem to become worth less since they can only travel a distance of three squares on a given move, whereas a bishop (traditionally considered

* Author's current address is: Department of Computer Science, Brandeis University, Waltham, Mass. 02254.

[1] Roughly speaking, a problem is PSPACE-hard if the existence of a polynomial time algorithm to solve it implies that any problem solvable with polynomial space is solvable with polynomial time. A problem is PSPACE-complete if it is both PSPACE-hard and solvable with polynomial space. See Aho, Hopcroft, and Ullman [1] or Garey and Johnson [6] for a discussion of these concepts.

0022-0000/83 $3.00

approximately equal to the knight in value) can travel up to N squares in one move. We avoid these problems and many others by defining a set of four axioms and showing that any generalization of chess satisfying these axioms yields PSPACE-completeness. Axioms 1, 2, and 3 state that any generalization of chess that contains at least 1 king, 1 queen, and a number of pawns that is polynomially related to the board size, qualifies for our result (in fact, at the end of Sect. 3, we note that these assumptions can be weakened). Axiom 4 states that the 50-move draw rule of standard chess[2] is generalized to a $P(N)$ draw rule, for some polynomial P. Without Axiom 4, the construction to be presented yields PSPACE-hardness rather than completeness. Although we have taken the view that exponentially long generalized chess games are not in the spirit of the 50-move draw rule, recently, Fraenkel and Lichtenstein [4] have shown that when exponentially long games are allowed, generalized chess is exponential-time complete. Their result does not apply to as wide a class of generalizations as covered by Axioms 1 through 3.

2. A Variant of Generalized Geography

Generalized Geography (GG) was used by the authors mentioned earlier to show checkers and GO PSPACE-hard. A variant of the GG problem will be used here. We start with a definition of GG. Note that for a node v in a directed graph, $IN(v)$ denotes the indegree of v and $OUT(v)$ the outdegree of v.

Definition. *Generalized Geography* (GG) is a two player game in which the players, called *White* and *Black*, move alternately, subject to the following conditions:

Board. Directed graph with a distinguished node called the *start node*, and with all nodes having nonzero outdegree.

Initial Configuration. White places a marker on the start node, and it is Black's turn.

Move Rule. A player may place a marker on any node that has an incoming edge, coming from the last node played.

Win Rule. A player loses by placing a marker on a node that already contains one.

The GG problem is: Given a directed graph G and a node s, does White having a winning strategy for playing GG on G with start node s? The *restricted* GG problem (RGG) is the GG problem with the added restriction that the graph be planar, bipartite, connected, contain no self loops, the start node s has $IN(s) = 0$ & $OUT(s) = 1$, and all other nodes v have $IN(v) = 1$ &' $OUT(v) = 2$, $IN(v) = 2$ &

[2] The 50-move draw rule of standard chess states that the game is declared a draw if 50 consecutive moves are made without a capture or a pawn move.

$OUT(v) = 1$, or $IN(v) = OUT(v) = 1$.[3] *Inverted* RGG is like RGG except that the win rule is changed to:

Win Rule. A player wins by placing a marker on a node that already contains one. ∎

Schaefer [10] shows GG PSPACE-complete via a redcution from the Quantified Boolean Formula problem (Meyer and Stockmeyer [9]). Lichtenstein and Sipser [8] present a clear presentation of how this is done[4], and extend this construction for RGG. Inverted RGG is easily shown PSPACE-complete by deleting the nodes from the back-edges in the construction presented in Lichtenstein and Sipser [8]. A variant of inverted RGG called *Array* GG will be used in this paper.

DEFINITION. A *grid graph* is a directed graph whose vertices are identified with distinct points of the planar grid, and whose edges are either horizontal or vertical line segments, where edges may intersect only at their endpoints. ∎

DEFINITION. *Array* GG (AGG) is inverted RGG, with the added restriction that the graph be a grid graph, where all nodes except the start node are of one of the four types (in any one of the four possible orientations) shown in Fig. 1a[5]. ∎

THEOREM 1. AGG is PSPACE-complete.

Proof. Given an instance G, s of RGG, we use a planarity algorithm to find a planar representation of G,[6] and then, working from the inside out, build an equivalent grid graph for G, s by considering G one region at a time. Figure 1b depicts how a region is "built on" to a partial embedding of G. In order to perform the transformation indicated in Fig. 1b, extra nodes may have to be added in order to turn corners. This is done by always adding nodes in pairs (so as to preserve the parity of the game). If there are more nodes to be put along the new squared-off path than there are corners on this path, the leftover nodes may be placed along the edges. Finally, before adding another region, local corrections of the type indicated in Fig. 1c may be required to ensure that nodes of degree 3 satisfy the JOIN or FORK format. The entire construction is given by Algorithm 1, which builds up the AGG problem G', s from the RGG problem G, s. Implicit in Algorithm 1 is that if more room is needed at a given point in the construction, the entire graph must be scaled

[3] It is easy to show the problem PSPACE-complete even if nodes v of the form $IN(v) = OUT(v) = 1$ are forbidden. We do not bother, since nodes of this form are required for the AGG problem to be defined shortly.

[4] Schaefer [10] and Lichtenstein and Sipser [8] use a slightly different definition of GG which is equivalent to the one presented here for the class of graphs containing no nodes of outdegree 0 (this must be true for RGG).

[5] Referring to Fig. 1a, in the FORK node, for example, we insist that not only is the node of indegree 1 and outdegree 2, but also that the incoming edge is perpendicular to the two outgoing edges.

[6] One such algorithm is described in detail in Hopcroft and Tarjan [7]. However, it is not actually necessary to use a planarity algorithm since the PSPACE-completeness proof for RGG is constructive.

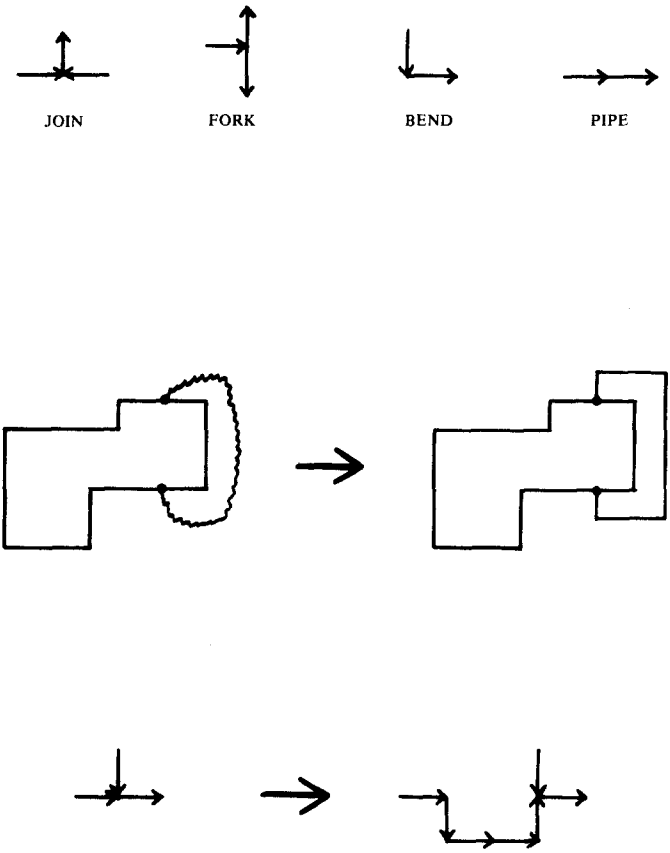


FIGURE 1

up in size. However, it is easy to show that even with this scaling and the introduction of new nodes in steps B and C, the size of G is polynomial in the size of G (and the algorithm runs in polynomial time and space). ∎

ALGORITHM 1

A. Label all of the regions and all of the edges of G "unmarked."
B. Mark a region of G and the edges on this region's perimeter.
C. Let G' be an embedding on a rectangle of the marked region's perimeter.
D. WHILE there is an unmarked region DO
    1. Fing an unmarked region R of G whose intersection with G' is a path.
    2. Add the perimeter of R not already in G' to G', as depicted in Fig. 1b.

3. Make any necessary local corrections, as depicted in Fig. 1c.
4. Mark R and the edges on its perimeter.

OD

E. Add the remaining unmarked edges of G to G'.

COROLLARY 1.1. AGG *is* PSPACE*-complete even if the* JOIN, FORK, *and* PIPE *nodes must be in the orientations shown in Fig.* 1a.

Proof. Figure 2a shows how to rotate a JOIN node 90 degrees. A FORK can be rotated in the same fashion (reverse arrows in Fig. 2a). Hence, by applying transformations of the type of Fig. 2a zero, one, two, or three times, all JOINS and FORKS can be put into the orientations shown in Fig. 1a. Figure 2b shows how to convert a vertical pipe to a horizontal pipe (reverse the arrows to go up instead of down). Finally, Fig. 2c shows how to change a horizontal pipe going to the left to one going

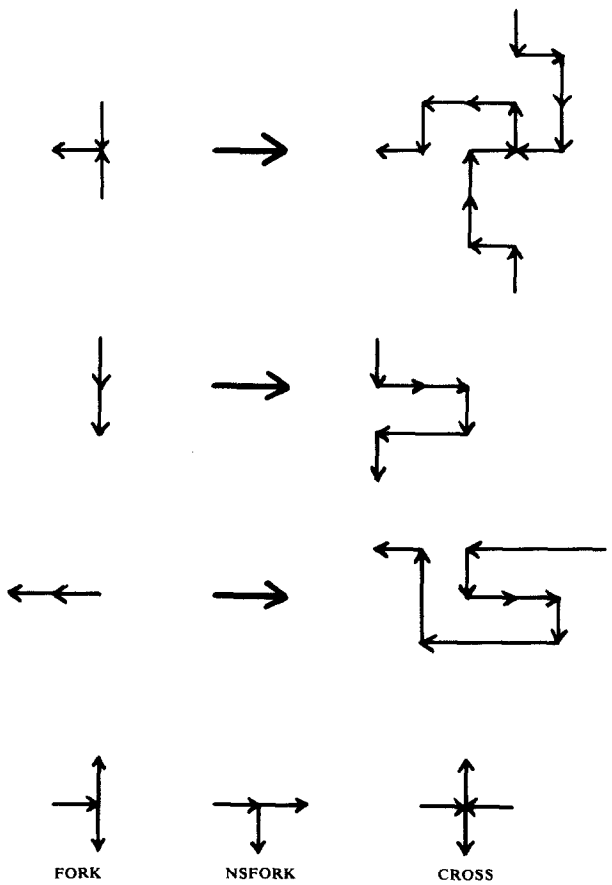


FIGURE 2

to the right. Note that as with Algorithm 1, this construction assumes that the graph is scaled up in size when necessary; the resulting size is polynomial in the initial size. ∎

The notion of AGG is fundamental to the construction presented in the next section. However, *nonstandard* AGG is what is actually used in the reduction, because this results in a slightly simpler proof.

DEFINITION. *Nonstandard* AGG is a two player game in which the players, called *White* and *Black*, move alternately subject to the following conditions:

Board. A grid graph where there is exactly one node with indegree 0 and outdegree 1 called the *START* node; nodes with indegree 1 and outdegree 0 are called *DEADEND* nodes; and all other nodes are one of the three types shown in Fig. 2d. The NSFORK and CROSS must be in the orientations shown in Fig. 2d (but the FORK can be in any of its for orientations). Furthermore, exactly 1 of the outgoing edges of every CROSS must be connected to a DEADEND.

Initial Configuration. Black places a marker on the start node, and it is White's turn.

Move Rule. A player may place a marker on any node that has an an incoming edge coming from the last node played, provided the last node played was visited for the first time. If, however, the last node played has been visited twice, then the player may only place a marker on an unvisited node that is adjacent to the last one played.

Win Rule. A player wins if he or she cannot move.

COROLLARY 1.2. *Nonstandard* AGG *is* PSPACE-*complete.*

Proof. First we observe that the only node that can be visited for a second time in a game of nonstandard AGG is the CROSS. Since every CROSS must have exactly one of its outgoing edges attached to a DEADEND, it follows that the only way that a player can win is to be stuck on a DEADEND. But this is tantamount to ruling that a player wins by visiting a CROSS for the second time. Hence, an instance of AGG where Black moves first (clearly, the PSPACE-completeness of AGG is independent of whether White or Black moves first) can be converted to an instance of nonstandard AGG as follows:

1. Replace each JOIN by a CROSS with the downward edge attached to a DEADEND.

2. Replace each PIPE by a NSFORK with the downward edge attached to a DEADEND.

3. Replace each BEND by the appropriate orientation of the FORK with the appropriate outgoing edge attached to a DEADEND. ∎

3. Main Result

As indicated in the introduction, we shall avoid the difficulty of defining a specific generalization of chess by proposing a weak set of axioms and proving PSPACE-completeness with respect to these axioms. At the end of this section, it is noted that a weaker set of axioms can be used.

Definition. *Generalized Chess* refers to any version of NxN chess that satisfies the following four axioms:

Axiom 1 (Initial Configuration). *There is a real number h (independent of N) such that at the start of the game, it is possible to draw a line horizontally across the center of the board so that all of White's pieces are on Whites's side of the line, all of Black's pieces are on Black's side, and no White piece is closer that N^h squares to any Black piece. Furthermore, for both Black and White, there is exactly one king, at least one queen, and at least N^h pawns.*

Axiom 2 (Move Rule). *White moves first, and pawns, queens, and kings move as in standard chess.*

Axiom 3 (Win Rule). *A player wins as in standard chess, that is, by checkmate of the opponent's king.*

Axiom 4 (Draw Rule). *The 50-move draw rule of standard chess is generalized to a $P(N)$ draw rule for some polynomial P.*

Clearly, in addition to the above axioms, there are a host of understood assumptions that forbid ridiculous rules such as "White is not allowed to checkmate Black."

Before proceeding, we present the following lemma, which ensures that the generalized chess positions used in our construction are, in fact, reachable from the starting position.

Lemma 1. *There is a constant $g > 0$, independent of N but dependent on the constant h of Axiom 1, such that from the starting position, it is possible via legal play to reach a position that has an arbitrary arrangement of pawns, queens, and the two kings inside the N^g by N^g center region of the board, and outside this center region there are exactly two pieces, one pawn of each color such that both pawns are at least N^g moves away from the edge of the board, from the center region, and from each other.*

Proof. First, as many queens of each color as needed may be created by each player advancing his or her rightmost pawns to become queens (while the opponent offers no resistance and allows pieces in the way to be captured). Queens and kings can easily move to any square of the board, and so the essence of the problem is to

show how to achieve an arbitrary position of pawns. This is done by shifting pawns from their original positions at the top and bottom of the board to the appropriate squares in the center; pawns can move vertically in the usual fashion and diagonally by capturing queens. This shifting may be done systematically by proceeding from left to right and filling in the N^g by N^g center region of the board one vertical column at a time. Since each time a pawn makes one square of horizontal progress by capturing a queen it also makes one square of vertical progress, sufficient vertical space must be between the pawns of the starting position and the N^g by N^g center region of the board; this is guaranteed by Axiom 1. In addition 1 guarantees that there is a sufficient number of pawns to carry out the above process. At this point, an arbitrary position has been achieved in the center region and all that is left is to remove all pieces outside this region except one pawn of each color. Each player can select a special queen and remove all of the opponent's pieces except the opponent's special queen and one pawn, which can be moved to be a distance of at least N^g away from anything (this must be possible if g is sufficiently small). Next, the two special queens can be captured by the opponent's pawn. One of the queens can choose or not choose to waste a move before moving to be captured by the opponent's pawn; this allows for it to be either player's turn once the process is completed. ∎

We shall henceforth refer to the N^g by N^g center region of the board guaranteed by Lemma 1 as the *virtual board.*

It should be noted hat it is possible to achieve arbitrary legal nonempty positions on the virtual board without leaving the two pawns outside, but this complicates the proof of Lemma 1. Since our construction will have the property that if a player can be checkmated in four moves if he or she does not respond, the two pawns outside the virtual board are of no consequence. In fact, it is only necessary to assume that they are at least five moves from the edge of the board (so they cannot possibly become queens in time to be of any use to a player).

The proof of Lemma 1 has the two players cooperating to achieve arbitrary legal positions. Whether the positions used in the construction to be presented bear any relation to positions that could be reasonably expected in "real play," is an issue that we do not address (and one that has not been addressed by the authors of other game results mentioned earlier).

THEOREM 2. *Given a position of generalized chess, it is* PSPACE*-complete to determine if White (or Black) has a winning strategy.*

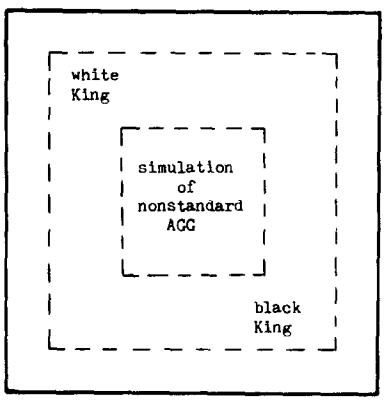
Proof. We construct a position consisting exclusively of pawns and queens that forces the players to simulate nonstandard AGG. The winner ends up with a "free queen," which may move to another part of the board and checkmate the opponent's king. The starting position is depicted in Fig. 3a. Note that for all figures showing chess positions, it is assumed that White's pawns move in the up direction and Black's pawns move in the down direction. The outer set of dashed lines in Fig. 3a indicate the virtual board guaranteed by Lemma 1; hence, reachability from the

starting position is guaranteed for the construction to be presented. The White and Black kings are in opposite corners of the board, surrounded tightly by pawns; Fig. 3b shows the position for the White king (the position for the Black king is symmetric).

CLAIM 1. *Placing a Black queen on either square x or y of Fig.* 3b *checkmates the White king. Furthermore, no four moves involving only the white pieces of Fig.* 3a *can change this.*

Proof of Claim 1. Clearly, the White king of Fig. 3b is trapped and is checkmated by a Black queen on square x or y. Furthermore, the diagonal isle containing the White king is too long for it to move out in four moves, and the walls of white pawns require at least four moves to make an opening. ∎

A consequence of Claim 1 is that it requires more than four moves for the white pieces of Fig. 3b to "untrap" the White king. Our construction will be such that if



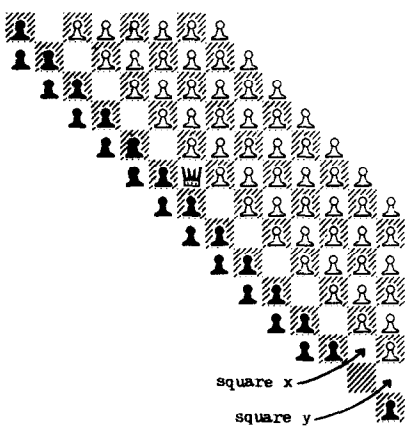


FIGURE 3

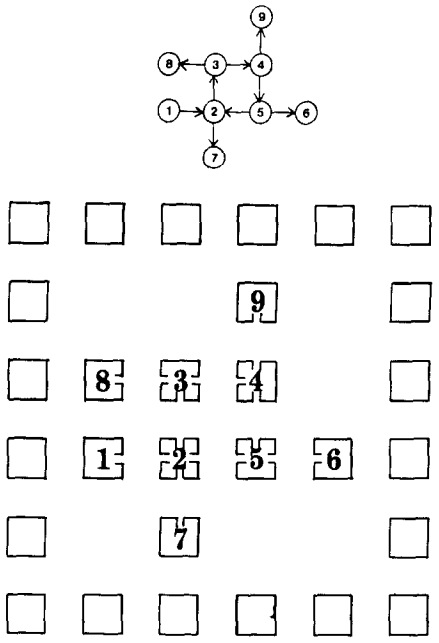


FIGURE 4

White (Black) should ever try to "waste" four moves to do this, the opponent will be able to get get a queen to either square x or y.

The center region shown inside the virtual board in Fig. 3a will be referred to the *simulation area*; that is, the kings are outside of the simulation area. The simulation of nonstandard AGG replaces each node of a graph by a corresponding chess position. Since an instance G, s of nonstandard AGG is played on a bipartite graph, we can divide the nodes of G into two sets; *White nodes*, where White enters and Black leaves, and *Black nodes*, where Black enters and White leaves. Figures 5 through 9 show the positions for the Black START, FORK, NSFORK, CROSS, and DEADEND nodes, respectively (positions for White nodes are symmetric[7]), using the following notation:

W denotes a White queen.

B denotes a Black queen.

P denotes a Black pawn.

The lines denote walls of pawns 2 thick. Walls of pawns are always separated by at least 2 empty squares from any piece inside.

The positions are shown with White's pawns moving in the up direction, and Black's pawns moving in the down direction.

[7] Except, of course, for the START node; there is exactly one START node, which is a Black node.

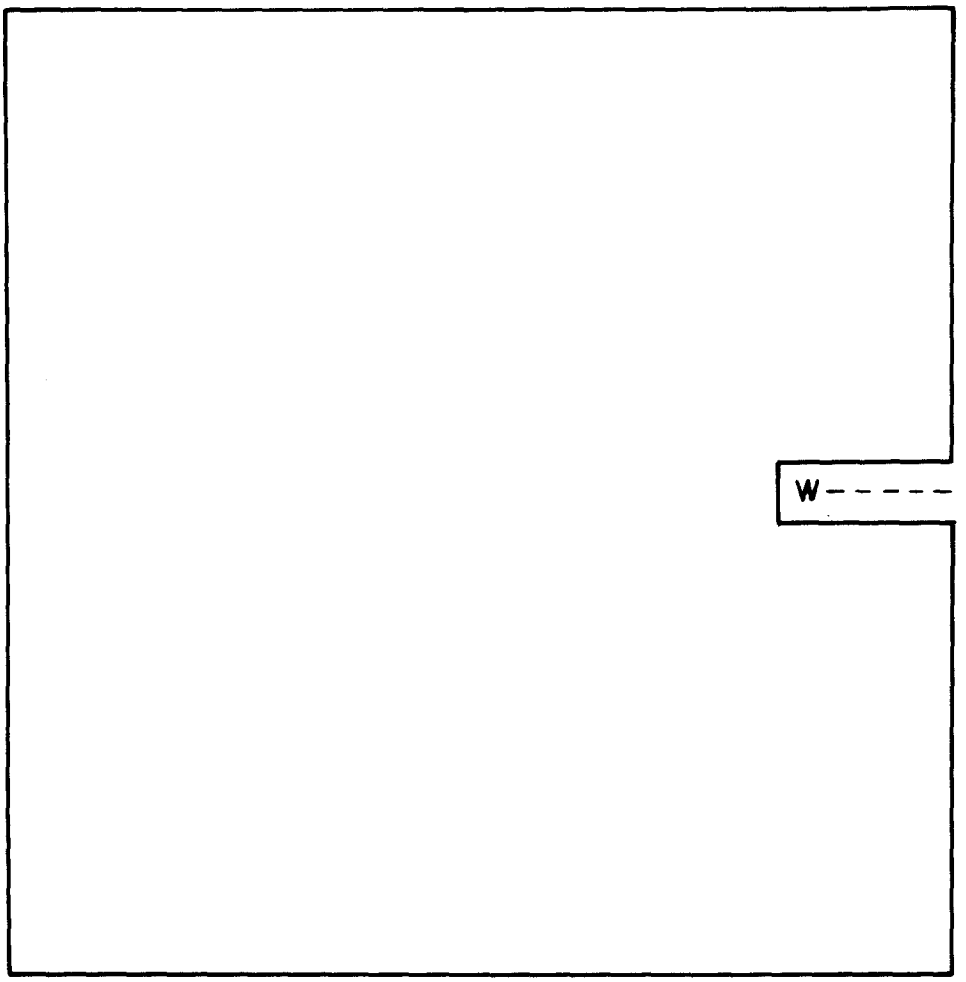


FIGURE 5

For the moment, it is not necessary to look closely at these figures, but only to note their shape. In particular, each position consists of a few pieces surrounded by walls of pawns having only a few openenings; these openings are called *ports*. Figure 5 (START) has one port, Fig. 6 (FORK) has three, Fig. 7 (NSFORK) has three, Fig. 8 (CROSS) has three, and Fig. 9 (DEADEND) has one. In addition, ports which are intended to be used as entrances for the simulation are called *entering ports*, and ports intended as exists are called *exiting ports*; the left port of Fig. 6, the left port of Fig. 7, the left and right ports of Fig. 8, and the port of Fig. 9 are entering ports, and all other ports in Figs. 5 through 9 are exiting ports. The walls of pawns in Fgs. 5 through 9 serve only to guide, contain, and surround positions, but never to come in contact with positions. In fact, the color of pawns for walls is unimportant.

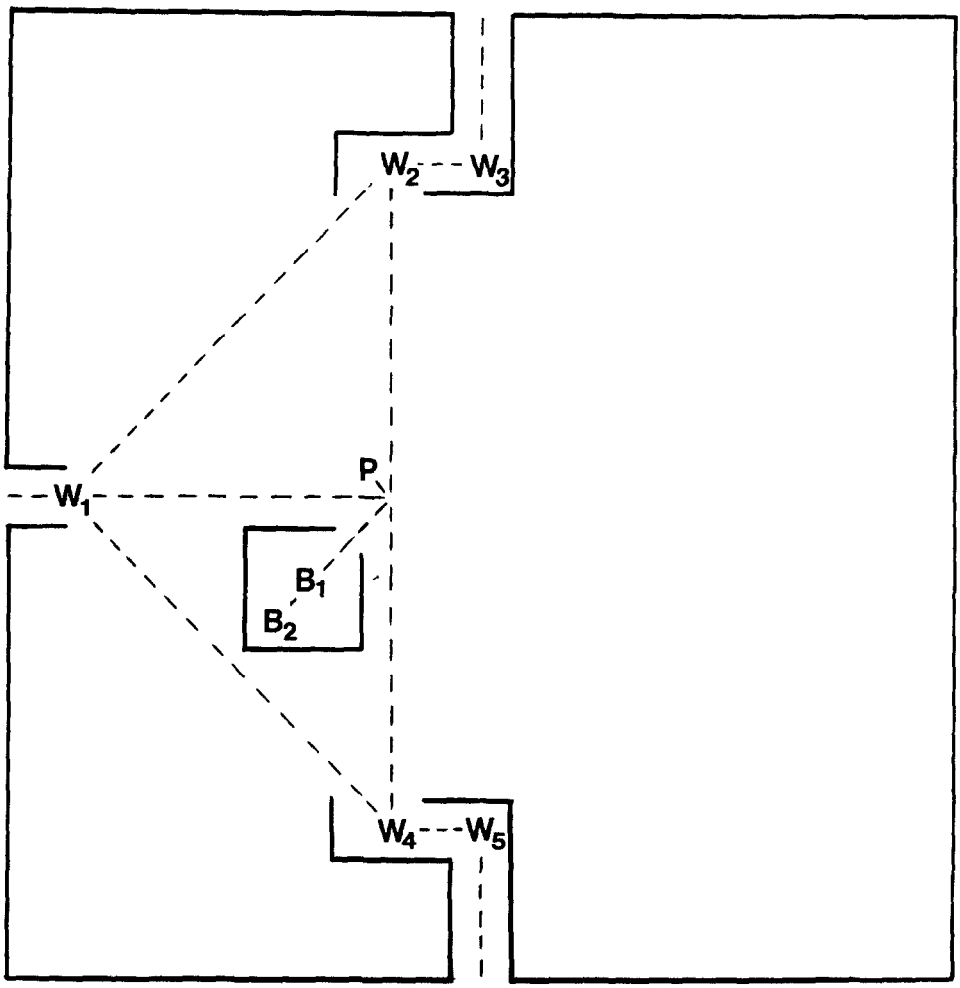


FIGURE 6

Figures 5 through 9 are drawn to the same scale, and it can be seen that they are all a k by k position of pieces for some constant k. Note that the ports in all of Figs. 5 through 9 line up exactly with either the vertical or horizontal center of the k by k square.

We can imagine a coarser grid where grid lines are separated by k of the "real" grid squares, and then map the nodes of an instance of AGG to the corresponding positions, placed on the coarse grid. To illustrate this, Fig. 4a is an instance of nonstandard AGG with a START node, a CROSS node, three FORK nodes, and three DEADEND nodes. Note that Fig. 4a is a bipartite graph, where odd numbered nodes are Black nodes and even numbered nodes are White nodes. Figure 4b depicts how the corresponding chess positions for Fig. 4a would be laid out in the center of

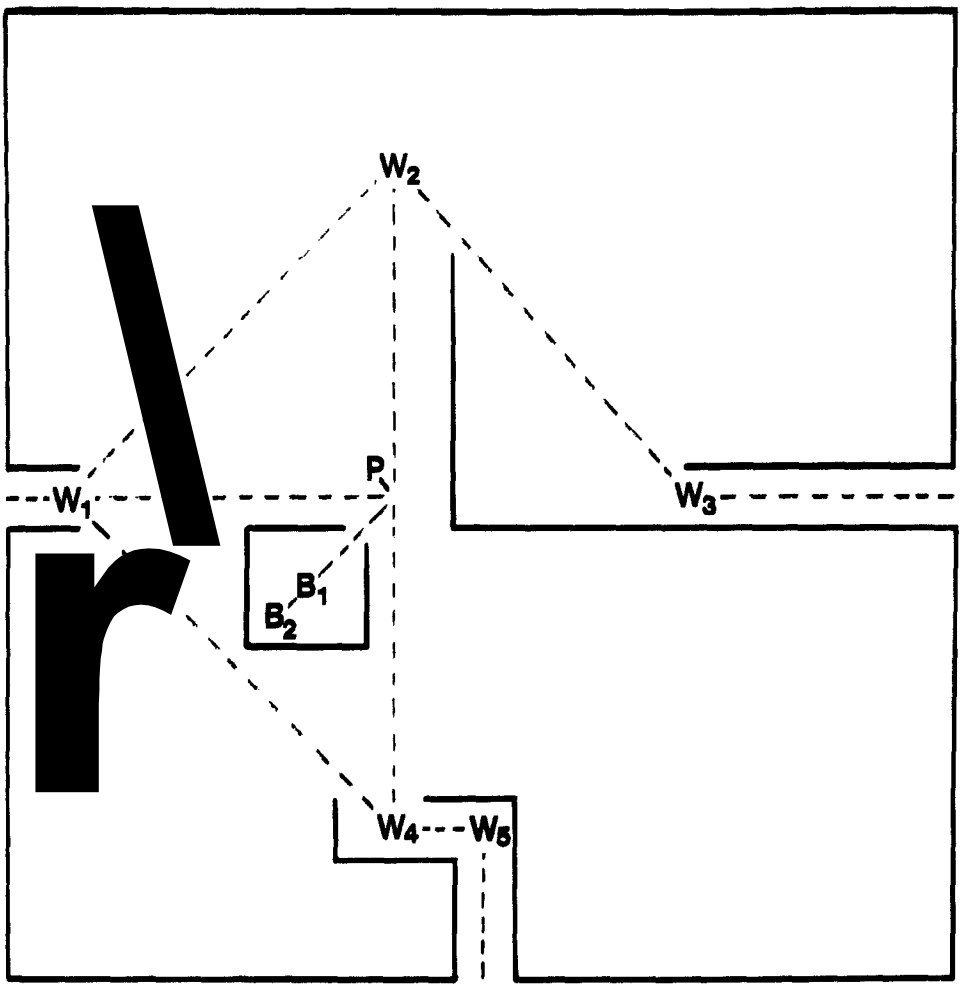


FIGURE 7

Fig. 3a. A key observation is that between every pair of horizontally adjacent nodes in Fig. 4b, there is a k-wide column of empty space extending vertically in both directions out to the dashed lines of Fig. 3a. A similar observation holds for every pair of vertically adjacent nodes in Fig. 4b. We shall refer to these empty k-wide columns as *escape isles* and to the k by k regions containing the positions for the nodes as *position boxes*. We say two position boxes are *adjacent* if the corresponding nodes in the game of nonstandard AGG are adjacent. Although the positions for the kings shown in Fig. 3b are outside the simulation area, for consistency, we shall assume that k is large enough so that Fig. 3b can be centered in a position box (with squares x and y lining up with the vertical and horizontal centers of the position box), and we shall refer to either of the two position boxes that contain the kings as a *king*

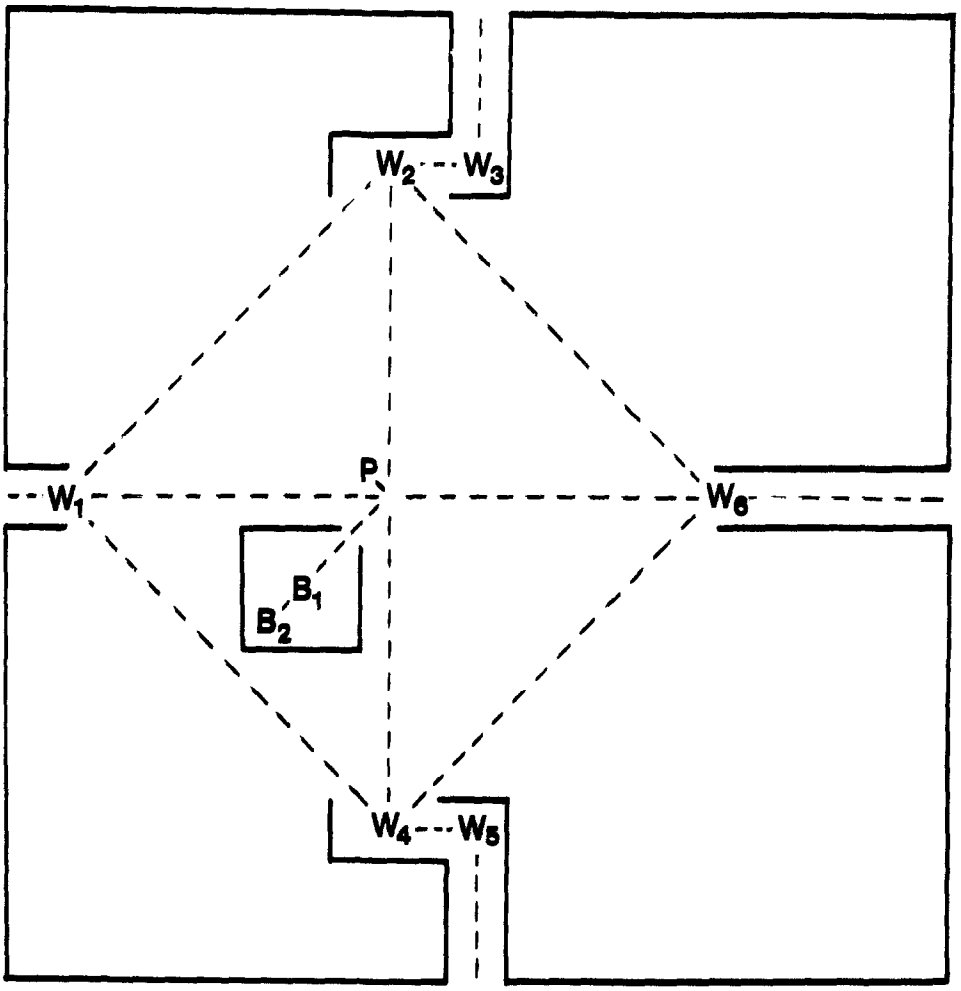


FIGURE 8

position box. During the course of play, we shall say a position box is *i-natal* if at most i of its pieces have moved from their original positions given by Figs. 3b and 5 through 9; a 0-natal position box is simply referred to as *natal.*

A position box that is completely filled with pawns is a *DUMMY position box.* Referring to Fig. 4b, it can be seen that the position boxes corresponding to the game of nonstandard AGG are surrounded by DUMMY position boxes. In general, we shall assume that this is always done. The surrounding wall of DUMMY position boxes insures that a queen can only leave the simulation area by either horizontally or vertically traversing an escape isle.

CLAIM 2. *The only way for a queen, in one move, to leave the simulation area*

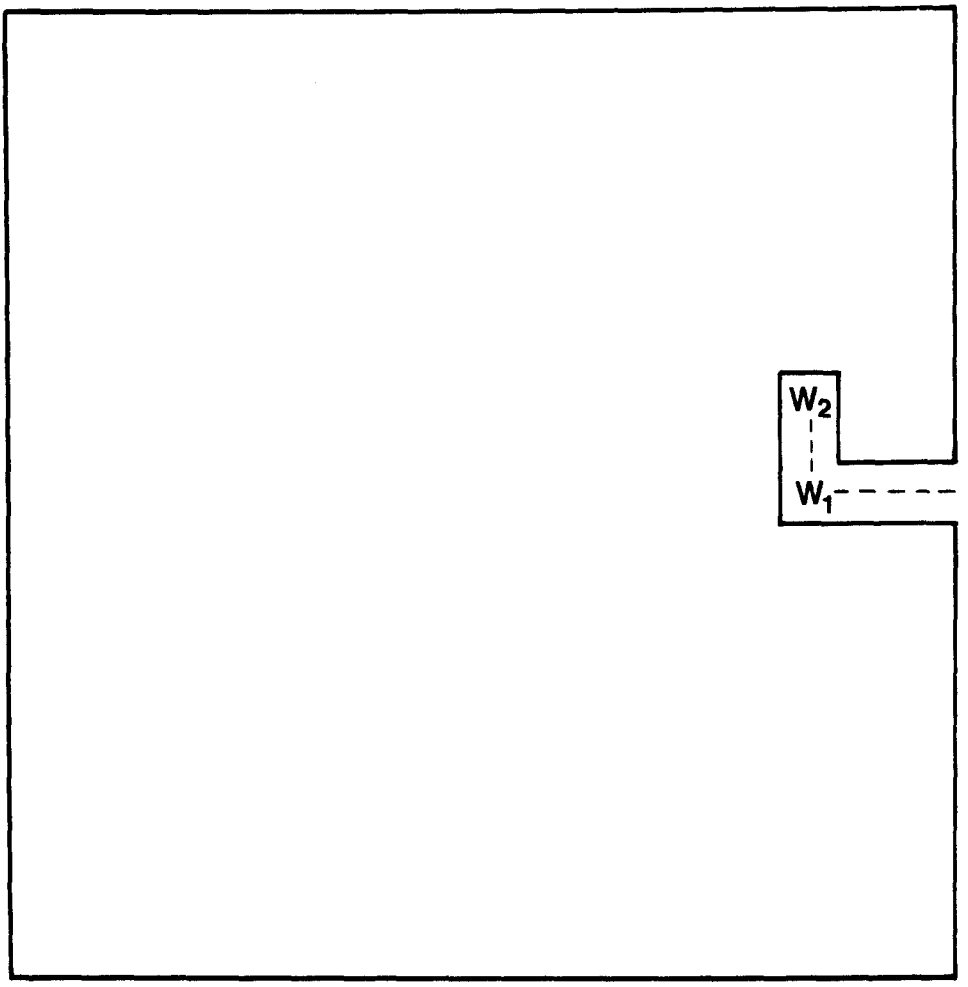


FIGURE 9

from a point inside the surrounding DUMMY position boxes is to move horizontally or vertically along an escape isle.

Proof of Claim 2. The only other move a queen can make besides horizontal and vertical, is diagonal. But since the DUMMY position boxes are k by k squares big but spaced only k squares apart, a diagonal move cannot possibly pass through the surrounding wall of DUMMY position boxes. ∎

Before proceeding, let us summarize where chess pieces are initially located for our construction. The portion outside the virtual board is essentially empty (it has exactly two pawns far away from the boarder of the real board or the virtual board). The only pieces inside the virtual board, but outside the simulation area, are pieces that

are part of the two king position boxes. Inside the simulation area, all pieces are part of some position box. There is a surrounding wall of DUMMY position boxes as depicted in Fig. 4b, and inside this wall, a simulation of nonstandard AGG is constructed with one START position box and some number of FORK, NSFORK, CROSS, and DEAD position boxes. As described earlier, the position boxes of the simulation area (including the surrounding wall of DUMMY position boxes) are separated by escape isles that lead horizontally and vertically out of the simulation area.

We have already defined the terms *virtual board*, *simulation area*, *escape isle*, *position box*, and *port*. We now define some terms pertaining specifically to queens:

For every port in Figs. 5 through 9, there is exactly one queen in the position that "guards" this port; we refer to these queens as *guarding queens*. For a given guarding queen, we refer to the guarding queen of the opposite color that is in line with it in the port to the adjacent position box as the *adjacent guarding queen*.

A queen of a given color is *free* if it is that color's turn to move, the queen is in an escape isle, and there is at most one queen of the opposite color that is outside a position box. Furthermore, if there is such a queen of the opposite color, then it must be inside the simulation area and in a different escape isle than the free queen.

A queen is *loose* if it is a guarding queen but is unprotected.

When it is White's turn to move, a position box is *benign* if its guarding queens are Black or if it satisfies the following conditions: If White moves a piece in this position box, then either Black will get a free queen or it will eventually become Black's turn, and White will have no queens in an escape isle. Benign for when it is Black's turn can be similarly defined (exchange "White" and "Black" in above).

If it is White's turn to move, a position box is *stable* if all of its guarding queens are black or the following conditions are satisfied:

(1) The king positions are natal.

(2) There is exactly one loose queen, which is White, and it is not part of this position box.

(3) This position box is benign.

(4) If White moves a piece that is part of this position box, then eventually, either Black can get a free queen or it becomes Black's turn with this position box again benign.

Stability when it is Black's turn is similarly defined (exchange "White" for "Black" in above).

Claim 3. Guarding queens are the only queens in a position box that can leave that position box in one move. All guarding queens for a given position box are the same color. Initially, every guarding queen except the one in the START position box is protected by at least one other queen in its position box; that is, initially, the guarding queen in the START position box is the only loose queen.

Proof of Claim 3. Easily checked. ∎

The key idea, to be described in detail shortly, is that at any stage in the simulation, there is exactly one loose queen, and the position box in which it is located corresponds to the node in the game of nonstandard AGG being simulated on which a marker was last placed. We shall refer to a position box as a *White position box* if its guarding queens are Black, and a *Black position box* if its guarding queens are White. Calling a position box Black, for example, reflects the fact that we are constructing this position box to simulate a Black node, where Black enters and White leaves.

Claim 4. A free queen can always checkmate the opponent's king, provided this king position box is 3-natal.

Proof of Claim 4. A free queen is always exactly two moves away from either square x or square y of Fig. 3b, because if it is in a horizontal escape isle, it can move horizontally and then vertically to square x, and if it is in a vertical escape isle, it can move vertically and then horizontally to square y. Furthermore, the opponent cannot prevent these two moves. This is because, by the definition of free, at most, one of the opponent's queens is in an escape isle; call this queen Q and the free queen Q_f. Without loss of generality assume that Q_f is in a horizontal escape isle. Let square z be the square (outside the simulation area) that Q_f must move horizontally to in order to line up with square x. By definition of a free queen, Q must be in the simulation area and in an escape isle different than Q_f. Hence, Q cannot be blocking the escape isle of Q_f, and Q cannot be covering square z (by Claim 2). Thus, on the first move, Q_f can certainly move to square z. Now it is the opponent's move. The opponent may be able to move Q vertically to cover square y (if Q is in a vertical escape isle), but this has no effect on Q_f moving to square x. The only other thing the opponent may be able to do (depending on the exact location of Q) to prevent Q_f from moving to square x on the next move, is to move Q horizontally to block Q_f; but then Q_f can capture Q, and since the opponent has no other queens outside of position boxes (by definition of free), Q_f can now move to square x on the next turn. Note that since we assumed the king box to be 3-natal, by Claim 1, this must result in checkmate (i.e., Black has had only one move to move something in the king position box). ∎

Claim 5. Suppose the king position boxes are natal, there is exactly one free queen which without loss of generality let us assume to be White, all position boxes are stable except the one containing a loose queen, it is White's turn, and neither player ever chooses to move a piece in the nonstable position box other than the loose queen. Then if White does not choose to move the loose queen to capture the adjacent black guarding queen, Black will get a free queen.

Proof of Claim 5. Call the loose queen W and let B denote the black guarding queen adjacent to W. If White moves some other piece, it must be in a stable position box. Hence, by the definition of stable, either Black will get a free queen or it will

become Black's turn to capture W with B, and all other position boxes are again benign. Since W was unprotected (by definition of loose), on the next move, White cannot prevent B from backing out into the escape isle. Furthermore, since all other position boxes are benign, White has no way to block B, and so on the next move, B becomes free. Hence, White must move W. Given this, it must be that W captures B. This is because the only other moves for W are to move to some square in line with B (either in W's port, in the escape isle, or in B's port), in which case B can capture W as before, or to move to some other part of its position box (or capture a pawn in one of the port walls), which allows B to move immediately out to the escape isle. ∎

Key Idea. Initially, the START position box contains the only loose queen and it is White's turn to move. In terms of the game of nonstandard AGG being simulated, White moving first in the simulation is equivalent to Black moving first in the game of nonstandard AGG, since the start position box is a black position box (i.e., we imagine that Black has occupied the START position box, and it is now White's turn to move). We shall consider each of the FORK, NSFORK, CROSS, and DEADEND position boxes and show that they are initially stable. Thus by Claim 4, White must move the queen in the START position box, and by Claim 5, this move must be to capture the guarding queen in the adjacent Black position box. In general, we shall show for each of the FORK, BEND, PIPE, and JOIN position boxes, not only are they initially stable, but moving a loose queen out of a port from a position box A to capture the guarding queen in an adjacent position box B, makes position box A stable and results in a sequence of moves leaving a loose queen in some other port of position box B. The only exception is when a player enters a DEADEND; in this case, the opponent gets a free queen and wins. Thus, the simualtion of nonstandard AGG has the property that there is always exactly one unstable position box, and flow proceeds from position box to position box (via loose queens) until a DEADEND is entered. Hence, White (Black) has a winning strategy if and only if this is true for the corresponding game of nonstandard AGG.

At the start of the simulation, by Claim 3, the queen in the START position box is the only loose queen, and by Claim 5, White must move to capture the adjacent Black queen. Note that Fig. 5 can be rotated to any orientation. Now, we must consider each of the FORK, NSFORK, CROSS, and DEADEND position boxes, and show that the position box is initially stable, and, except for the DEADEND position boxes, if the current loose queen is in a port adjacent to an entering port of the position box, then a sequence of moves must take place which causes the current loose queen to be in an exiting port of this node. For a DEADEND position box, we show that a player moving into one is sure to lose. Figures 6 through 9 are the Black position boxes; hence, our arguments will be from the point of view that the current loose queen is Black and it is Black's turn. The White position boxes and the corresponding arguments are symmetric. Note also that initially, the king position boxes are natal, and for every position box we shall verify that no player has time to move a piece in a king position box, and so the king position boxes must remain natal (until, possibly, a player is in the process of getting a free queen).

Consider the Black FORK position box shown in Fig. 6. Note that Fig. 6 is the "west" orientation; we consider the other three orientations later. Referring to Fig. 6, note that P is positioned so that it threatens the square covered by W_1, W_2, W_4, and B_1. Although Fig. 6 is a Black position box[8] and we are primarily concerned with the case when it is Black's turn, and the current (Black) loose queen is in the port adjacent to the entering port to the FORK node, we start by checking that the FORK node satisfies our stability assumptions when the current loose queen is not adjacent to it. To do this using Fig. 6, let us for the moment assume that it is White's turn (and the current loose queen is White). As discussed earlier, we can assume that king position boxes are natal. Figure 6 is clearly benign, since moving any of the White queens to a point inside the position box or moving W_1, W_3, or W_5 to a point inside an adjacent port leaves no White queen in an escape isle, and moving W_1, W_3, or W_5 into an escape isle allows the adjacent Guarding queen to capture it and become free on the next move. Thus, we are left with verifying Condition (4) of the stability definition. Note that Condition (4) of the stability condition is tantamount to asserting that moving a White queen in a Black node other than the one containing the current loose queen is fruitless. To be completely thorough, we should also verify that moving a White queen in a White position box other than the one adjacent to be loose queen (this node will be treated in detail shortly) is also fruitless, but this will easily follow from the following arguments, since the guarding queens for a White position box are all Black. We have already noted that moving W_1, W_3, or W_5 into an escape isle results in Black getting a free queen. If W_1, W_3, or W_5 move to capture an adjacent guarding queen, then Black can recapture with the queen protecting that Guarding queen, call this queen B, and now B is threatening to move out into the escape isle. Now White has to move W_2 or W_4 to the vacant port to cover B. At this point, it is again Black's turn, and it can be checked that this node is again benign. It is also not hard to check that the nodes adjacent to this node must also again be benign. Thus, trying to move W_1, W_3, or W_5 out of the position box is fruitless for White. The only other nontrivial move for White is to move W_2 or W_4 to the center square, but then B_1 moves up to capture, and again, White's position has degraded.

The above paragraph examined the structure of the FORK node when it is not adjacent to the current loose queen; since Fig. 6 shows the position box for a Black FORK node, we assumed that it was White's turn. Let us now assume that it is Black's turn, and let B_0 denote the current loose queen, which is adjacent to W_1. We start by simply listing the sequence of moves that must occur if B_0 is the current loose queen.

(A) B_0 captures W_1.

(B) W_2 or W_4 captures B_0; without loss of generality, assume W_2 captures B_0.

[8] Recall that a White position box is one with Black guarding queens (i.e., White enters and Black leaves), and a Black position box is one with White guarding queens (i.e., Black enters and White leaves).

(C) B_1 moves up to cover W_2.

(D) W_3 becomes the current loose queen.

We now consider why the above sequence of moves must take place. By Claim 4, Black's only option is to capture W_1 with B_0 or to move P, B_1, or B_2. If P is moved, then by the same reasoning as used for the proof of Claim 4, White will get a free queen. B_2 cannot move out of its containing box in one move unless B_1 does. The only productive move for B_1 is to move up to cover W_1. But now W_2 captures B_1 (and again, White is threatening to capture B_0 with W_1). At this point, there are three cases:

Case 1. B_0 captures W_1: Then W_2 captures B_0. Now, White is sure to get a free queen; trying to take W_3 from an adjacent position box is fruitless (since W_2 will be one move ahead), and moving B_2 is fruitless, since W_4 captures B_2 (and W_2 is still one move ahead of a Black queen capturing W_3 or W_5).

Case 2. P captures W_2: Now, P is blocking B_1 from moving up, and so W_1 can go ahead and capture B_0.

Case 3. B_2 captures W_2: Then W_4 captures B_2, and we can apply the same reasoning as Case 1.

Thus, referring to (A), it must be that B_0 captures W_1. At this point, W_2(or W_4) must capture B_0 or B_0 leaves the way it came to become free. Now, B_1 must move up to prevent W_2 from becoming free on the next move. W_3(or W_5) is now the current loose queen.

We must now check that the position box again satisfies our stability assumptions. It doesn't pay for White to capture B_1 with W_2 or W_4 because if W_2 captures B_1, B_2 recaptures W_2, W_4 must capture B_2, P captures W_4, and White has gained nothing but has exposed W_5. If W_4 captures B_1, B_2 captures W_4, W_2 captures B_2 (if not, by stability, B_2 will eventually capture W_2 to become free), P captures W_2, and again White has gained nothing. Hence, after (D), the stability assumption is preserved.

The "east" orientation of the FORK position box is just the mirror image of Fig. 6, and all reasoning is the same. To obtain the "north" orientation, Fig. 6 can be rotated 90 degrees clockwise; since P is a Black pawn, it will still be covering the center square. For the "south" orientation, Fig. 6 can be rotated 90 degrees counter-clockwise, and then P can be reflected across the center square to obtain a completely equivalent position to Fig. 6.

The NSFORK position box can be obtained by transforming the FORK position box. It can be checked that Fig. 7 is functionally identical to Fig. 6, and so the same argument as for the FORK can be used. By definition of nonstandard AGG, Fig. 7 is the only orientation of the NSFORK needed.

Let us now consider the Black CROSS position box, shown in Fig. 8. Initial stability of the CROSS can be verified in a fashion similar to the FORK, and we leave this to the reader. Let us now assume that it is Black's turn, and let B_0 be the current loose queen, which without loss of generality, assume is adjacent to W_1

(adjacent to W_6 is symmetric). Also, let B_3 denote the guarding queen adjacent to W_6. Again, we start by listing the sequence of moves that most occur. Note that P is positioned so that it covers the center square.

(A) B_0 captures W_1.

(B) W_2 or W_4 captures B_0; without loss of generality, assume W_2 captures B_0.

(C) B_1 moves up to cover W_2.

(D) W_3 becomes the current loose queen.

In addition, if subsequently B_3 becomes the current loose queen then:

(E) B_3 captures W_6.

(F) W_4 captures B_3.

(G) B_1 captures W_2 or W_4; without loss of generality, assume B_1 captures W_2.

(H) W_4 captures B_1.

(I) B_2 moves up to cover W_4.

(J) W_4 captures B_2.

(K) P captures W_4.

(L) W_5 becomes the current loose queen (and White is sure to lose because the position box adjacent to W_5 is a DEADEND).

As with the FORK position box, we must now verify that the above sequence of moves must occur. Although the reasoning for steps (A) through (D), which are idential to those for the FORK, is a bit more complex than that for the FORK, it is similar in flavor. By Claim 5, Black's only option is to capture W_1 with B_0, or to move one of the pieces P, B_1, or B_2. Using reasoning similar to what was used for the FORK, it can be shown that moving P, B_1, or B_2 is fruitless. Hence, referring to (A), B_0 must capture W_1. At this point, White must do something or else B_0 moves back out, W_6 captures B_0, and B_3 captures W_6 to become free. If W_6 captures B_0, B_1 moves up (threatening to move B_3 out), and now a trade off follows that leaves both W_3 and W_5 exposed. Thus, referring to (B), White must capture B_0 with W_2 or W_4. Without loss of generality, assume White chooses W_2. Referring to (C), B_1 must move up to prevent W_2 from escaping. At this point, White has no profitable moves, and referring to (D), W_3 becomes the current loose queen.

We now check that this position box again satisfies our stability assumptions. Figure 10 shows what Fig. 9 looks like after it has been transformed by the moves (A) through (D) described above, and after the loose queen W_3 has left. We must show that if it is White's turn to move and the current loose (White) queen is not adjacent to W_6, then moving W_2, W_4, W_5, or W_6 is fruitless. If W_2 moves out into the escape isle, then B_3 captures W_6 to become free on the next move; there are no other nontrivial moves for W_2. The reasoning for W_6 is similar. If W_5 moves out into the escape isle, it may be captured by the adjacent guarding queen, which then becomes free on the next move. Hence, the only nontrivial move for W_5 is to capture

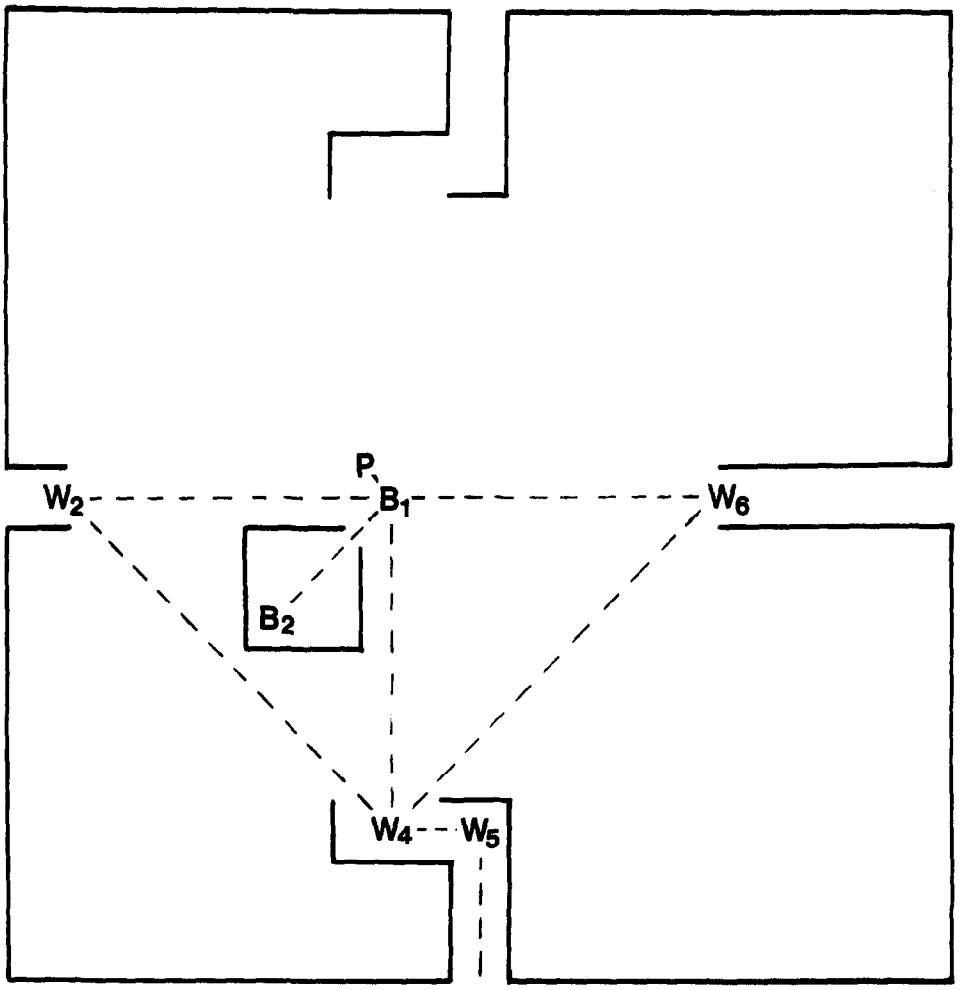


FIGURE 10

the adjacent guarding queen (which, incidently, must be part of a DEADEND). But then the queen protecting this guarding queen, call if B, can recapture. Now, W_4 must move to the port to cover, leaving W_2 and W_6 unprotected (allowing B_3 to capture W_6, W_2 captures B_1, B_2 captures W_2, and B_2 escapes). Given that W_2, W_5, and W_6 cannot move, W_4 has no nontrivial moves. Hence, this node is again stable.

Now, referring to Fig. 10, assume that (later on) B_3 becomes the current loose queen (remember that B_3 denotes the guarding queen in the adjacent position box to the right). It is sufficient to show that White's moves (F), (H), and (J) are forced, given that Black makes moves (E), (G), (I), and (K); this is because these moves result in W_5 becoming the current loose queen, a winning strategy for Black (since W_5 is adjacent to a DEADEND). Given that B_3 chooses to capture W_6, Black is

now threatening to move B_3 back out into the escape isle, and White has only three options: W_2 captures B_1, W_4 captures B_1, or W_4 captures B_3. If W_2 captures B_1, then B_2 captures W_2, and now Black is sure to get a free queen (W_4 captures B_2 or B_3 just delays this one move). Similarly, if W_4 captures B_1, B_2 captures W_4 and Black is again sure to get a free queen (W_2 captures B_1 only delays this one move). Hence, W_4 must capture B_3. B_1 can now capture W_2 or W_4. Without loss of generality assume B_1 captures W_2. Then W_4 must capture B_1 (to prevent B_1 from capturing W_4 on the next move) and now B_2 can move up to cover W_4 (to prevent W_4 from leaving). W_4 is now in danger of being captured by B_2(and then B_2 would be free to leave). W_5 cannot move to protect W_2 since then the guarding queen adjacent to W_5 could move out into the escape isle and become free on the next move. Hence W_4 must capture B_2 and now, P can capture W_4(to prevent W_4 from leaving). There are now no queens left in this node except W_5, which is the current loose queen.

We now consider the DEADEND position box, shown in Fig. 9. Assume that B_0 is the current loose queen, which is adjacent to W_1. Since Fig. 9 contains no Black queen, by Claim 4, Black's only option is to capture W_1 with B_0. But now W_2 captures B_0 and there is no way for Black to prevent W_2 from becoming free on the next move. ∎

It is unikely that anyone would disagree that any reasonable generalization of chess should have at least one piece that moves like the standard queen. However, it could be argued that the king and/or pawn moves should be more powerful. We close by noting that with reasonable generalizations of the pawn and/or king moves, the construction we have presented goes through essentially unchanged.

THEOREM 3. *For any function $f(N)$ growing slower than N^{1-h} for some $h > 0$, Theorem 2 holds for any generalizations of the king and pawn moves that satisfy:*

King: The king cannot move a distance greater that $f(N)$ squares in a single move.

Pawn: A pawn may move and capture at most a distance of $f(N)$ squares in the forward direction (diagonally or straight), and is guranteed to capture on at least one square in the two forward diagonal directions.

Proof. Referring to the construction of Theorem 2, the pawns around the kings move away from the kings; thus there is no problem here except that the walls may have to be made at most $f(N)$ longer and thicker, so that the pawns close to the king are protected and the kings cannot escape. As for the barriers around the queens, it is only necessary to move them so that no queen is closer than $2f(N)$ squares to a pawn. Also, the ports to the queen configurations can be lengthened by at most $2f(N)$ to ensure that a queen cannot escape diagonally. The two active pawns used in the position box simulations may be placed appropriately. It should be noted that this construction yields a polynomial relationship between the board size and the size of the AGG game. ∎

The above theorem addresses Axiom 2 of Generalized Chess. It is not hard to see that Axiom 1 can also be greatly generalized. In addition, theorems can be stated with regard to multiple kings, nonsquare boards, etc. All such results may exploit the fact that, in the construction of Theorem 2, pawns (with four exceptions) are used only as barriers, and the kings are outside the simulation area.

4. Conclusion

An interesting direction for future research would be to unify the results of this paper and those for other games such as checkers and GO. Many two-person information-perfect board games can be defined in a fashion similar to that of Definition 4 (i.e., an initial configuration rule, move rule, win rule, and possibly a draw rule). A formalization of this concept leading to a general complexity theorem that subsumed chess, GO, checkers, and other games as special cases would be significant.

References

1. A. V. Aho, J. E. Hopcroft, and J. D. Ullman, "The Design and Analysis of Computer Algorithms," 2nd ed., Addison–Wesley, Reading, Mass., 1974.
2. A. K. Chandra and L. J. Stockmeyer, Alternation, *in* "*Proceedings*, 17 Annual IEEE Symposium on Foundations of Computer Science, Long Beach, California," pp. 98–108, 1976.
3. S. Even and R. E. Tarjan, A combinatorial game which is complete in polynomial space, *J. Assoc. Comput. Mach.* **23** (1976): 710–719.
4. A. S. Fraenkel and D. Lichtenstein, Computing a perfect strategy for *NxN* chess requires time exponential in *N*, *J. Combin. Theory Ser. A* **31** (1981): 199–214.
5. A. S. Fraenkel, M. R. Garey, D. S. Johnson, T. J. Schaefer, and Y. Yesha, The complexity of checkers on an *NxN* Board—Preliminary Report, *in* "*Proceedings*, 19 Annual IEEE Symposium on Foundations of Computer Science, Ann Arbor," Michigan, pp. 55–64, 1978.
6. M. R. Garey and D. S. Johnson, "Computers and Intractability," W. H. Freeman, San Francisco, 1979.
7. J. E. Hopcroft and R. E. Tarjan, Efficient planarity testing, *J. Assoc. Comp. Mach.* **21** (1974): 549–568.
8. D. Lichtenstein and M. Sipser, GO is PSPACE-Hard, *Proceedings*, 19 Annual IEE Symposium on Foundations of Computer Science, Ann Arbor, Michigan, 1978, pp. 48–54.
9. A. R. Meyer and L. J. Stockmeyer, Word problems requiring exponential time, *in* "*Proceedings*, Fifth Annual ACM Symposium on Theory of Computing, Austin, Texas," pp. 1–9, 1973.
10. T. J. Schaefer, Complexity of decision problems based on finite two-person perfect-information games, *in* "*Proceedings*, Eighth Annual ACM Symposium on Theory of Computing, Hershey, Pennsylvania," pp. 41–46, 1976.